System Specification

Commanding and Command Processor Thread, Thor DP1

Checkout and Launch Control System (CLCS)

84K00302-025

Commanding and Command Processor Thread

Assessment

September 29, 1997

Version 1.0

Commanding and Command Processor Thread
Table of Contents

Commanding and Command Processor Thread
Table of Figures

Commanding and Command Processor Thread
Table of Tables

Assessment Team

	CI Represented	E-Mail Address	Phone
Walter Clavette	Command Support	ClaveWP@kscgws00.ksc.nasa.gov	1-2213
Tom Jamieson	System Engineering	JamieTJ@kscgws00.ksc.nasa.gov	1-2263
Gary Hrezo	System Viewers	HrezoGW@kscgws00.ksc.nasa.gov	1-7445
Stan Siemienski		SiemiSJ@kscgws00.ksc.nasa.gov	1-2206
Ray Bessenaire	TAS	BesseRG@kscgws00.ksc.nasa.gov	
Brian Hooker	Command Support	HookeBC@kscgws00.ksc.nasa.gov	1-7443
Robert Owens	Command Support	OwensRL@kscgws00.ksc.nasa.gov	1-7384
Robert Raymond	Command Support		
Mike Krzos	Command Support	KrzosMJ@kscgws00.ksc.nasa.gov	
Jack Blackledge	Thread Lead	BlackJW@kscgws00.ksc.nasa.gov	1-2272
Julia Samson	Application Services	SamsoJH@kscgws00.ksc.nasa.gov	1-2212
Debbie Lee	User Applications	LeeDA@kscgws00.ksc.nasa.gov	
Mike Lunceford	GSE Gateway	LunceJM@kscgws00.ksc.nasa.gov	1-2294
Scott Cummins	LDB Gateway		1-2291
Charla King	DBSafe	KingCG@kscgws00.ksc.nasa.gov	1-7587
Rich Ikerd	User Applications	IkerdRG@kscgws00.ksc.nasa.gov	1-7541
Chau Le	PCM Gateway	LeCB@kscgws00.ksc.nasa.gov	1-7681
Dave Harrington	Integration & Test	HarriDE@kscgws00.ksc.nasa.gov	1-7370
Robert Pierce	Engineering	PiercPM@kscgws00.ksc.nasa.gov	
Wesley Tharpe	System Test	TharpeWJ@kscgws00.ksc.nasa.gov	1-7373
John Copella	System Engineering		1-2237

Table 1 Assessment Team

1. Introduction

1.1 Commanding and Commanding Processor Thread Overview.

The Commanding and Command Processor thread supports commanding by cursor control, manual input, Test Application Scripts, and End item Managers. In addition, it provides the basic command processor GUI and command structure for CLCS. The interface for all commands will be implemented, however, not all commands will be implemented.

1.2 Commanding and Command Processor Thread Concept

RTPS commands are directed to end item function designators, RTPS system functions, and applications software. Figure 1 illustrates the RTPS conceptual model for commands issued to function designators and RTPS system functions. Function designator commands and RTPS system commands originate at the Command and Control Workstation from user displays, the Command Processor, or command scripts. End Item Managers in the Command and Control Processor may also issue function designator commands and some RTPS system commands. In both systems, the Applications Services provide the interface for commands to be transmitted to the Command Manager. The Command Manager performs authentication functions appropriate for the subsystem location. Command Management in the CC W/S uses Inter-process communications to forward commands to the CCP Command Management. In the CCP, Command Management performs additional authentication, and, for Function Designator commands, Prerequisite Control Logic checks are performed. Authorized commands are then transmitted to the appropriate Gateway subsystem or to the appropriate RTPS system function.

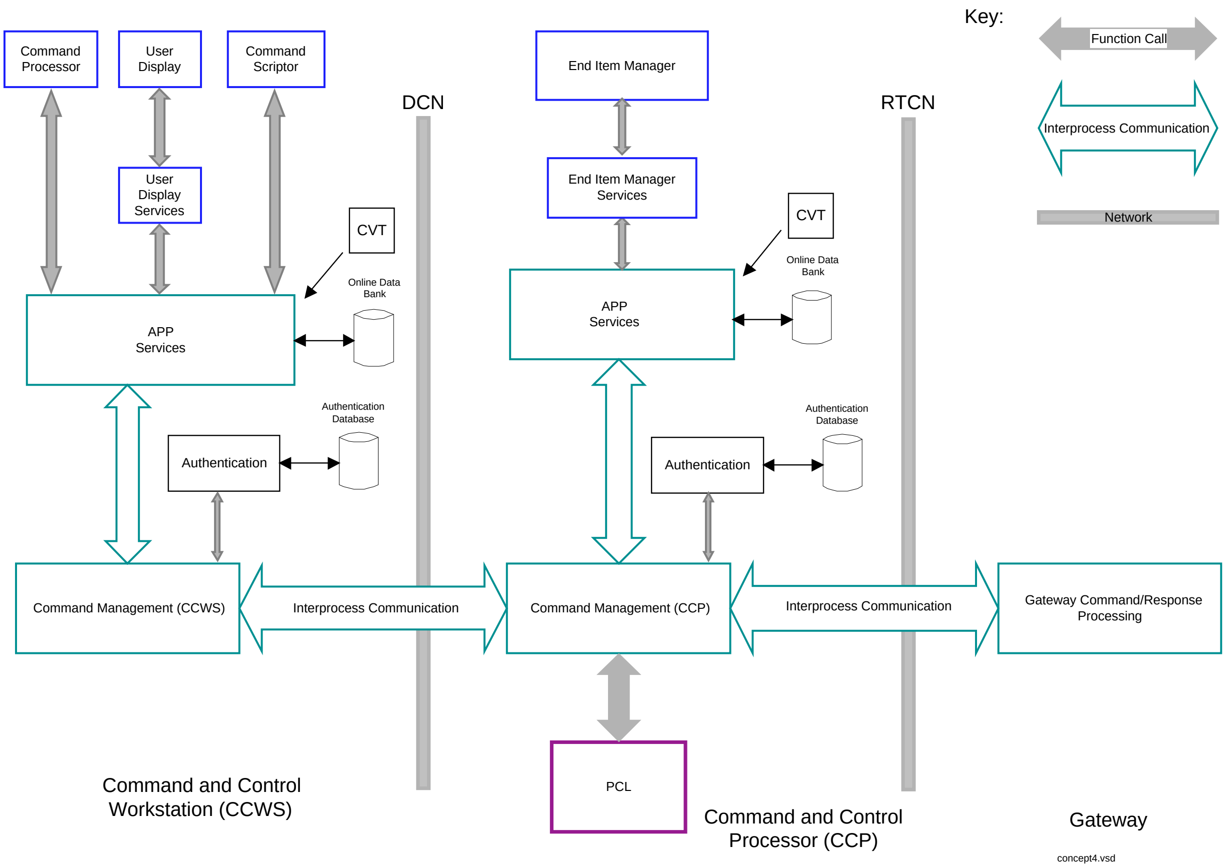

Figure 1 Commanding and Command Processor Thread Concept - 1

Figure 2 illustrates the RTPS conceptual model for application software commanding. Commands from the user interface in the CC W/S are transmitted to applications software in the CCP via CORBA (Common Object Requset Broker Architecture). Command Management in the CCP provides a CORBA filter that performs appropriate authentication on applications commands. Authorized commands are then transmitted to the targeted application (End Item Manager). This filter is also used to authenticate End Item Manager to End Item Manager commands.

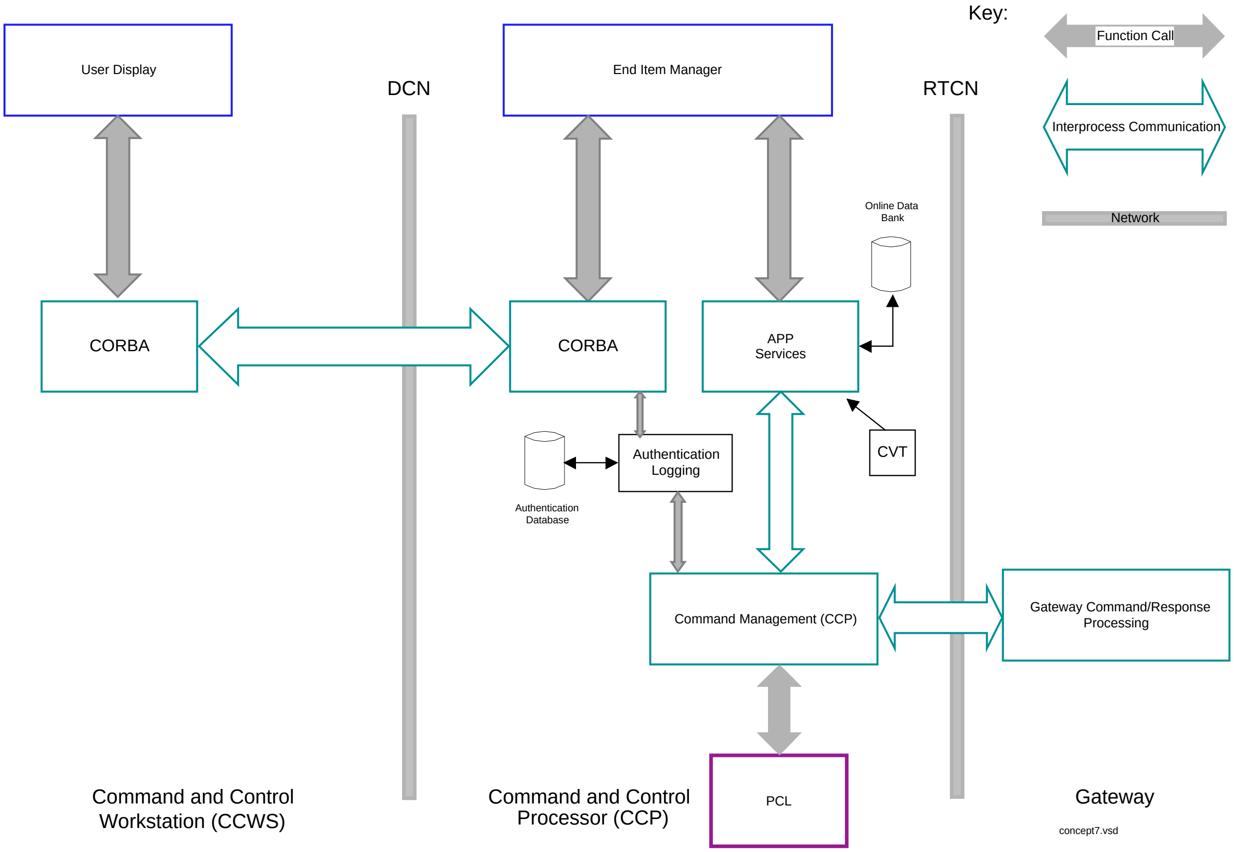

Figure 2 Commanding and Command Processor Thread Concept - 2

1.3 Commanding and Command Processor Thread Specification

1.3.1 Statement of Work

The Thor Commanding and Command Processor Thread statement of work from the Thor Delivery Document.

Command Management

- Add mode control to FD commands (~~auto, direct~~, override~~, bypass~~)
- Add platform routing to commands
- Add priority queue ***in LDB Gateway*** for ***LDB*** commands
- Add initial commanding to support End Item Manager control
- Add routing of FD Commands to Prerequisite Control Logic
- ~~Add routing of FD Commands to End Item Manager~~
- Provide command interface from End Item Managers and Test Application Scripts
- Provide initial ***EIM*** application to ***EIM*** application commands

- Provide initial Launch Data Bus FD commands (AS, DS)
- Provide initial Launch Data Bus memory read write commands ***(GMEM Read, GMEM Write, GPCC LDBC)***
- Provide Table Management commands for Ground Support Equipment Gateway
- Provide a user display capability to demonstrated FD's related command and modes.
- ***Provide prototype of $CMD function for CLCS***
- ***Define and prototype the design for command authentication processing.***
- ~~Provide object based communication between user application using CORBA's Interface Definition Language for interface specification. Utilize CLCS Reliable Messages packets for the transport mechanism.~~

Command Processor

- Evaluate command interface for use of code and or methods in CLCS
  - Launch Team Training System completion editor.
  - Launch Team Training System command windows.
  - TCMS
  - CCMS 1
  - JSC
  - Others.
- Develop the list of command functions using as a basis but not limited to the SLS table.
- Develop a matrix of command methods (list select, command line, graphical, menus) and map them to command functions.
- Prototype various methods ***of commanding*** and receive user feedback.
- Provide specific support for command GUIs needed by Commanding Phase 2 Thread
- Implement a set of GUIs to provide most of the command functions needed in CLCS

Performance Requirement from SLS

- The system shall support executing a manual command in less than one second from human execution to HIM output.
- The system shall provide the capability to issue commands from a single test (control) application to support every Launch Data Bus command opportunity (120 milliseconds).
- RTPS shall be able to support full Uplink command rates on the following links:
  - GSE - 500/second.

1.3.2 Requirements

These requirements have been extracted from the SLS Rev Basic, June 26,1997. This is more than required for Thor. These requirements will for the basis for developing the Thor functional requirements. All post-Thor requirements are italicized.

(SLS - 2.1.1.1.1) *The RTPS shall meet the requirements allocated to the LPS/LDB interface specified in:*

1. *SS-P-0002-150, Space Shuttle LDB Software Interface Requirements*
2. *ICD-2-0A003, Section 3, Flight Vehicle/LPS Computational Systems Interface*

(SLS - 2.1.1.2) The RTPS shall provide the capability, in GPC Mode, to issue commands and to receive measurement data, via the GPCs, from:

1. Orbiter Multiplexers/Demultiplexers (MDM) (i.e., Flight Critical, *Payload, Flex, SCA, and Command Decoders)*

2. *Master Events Controller (MEC)*
3. *Pulse Coded Modulation Master Units (PCMMU)*
4. *Mass Memory Units (MMU)*
5. *SSME Controllers*
6. *Solid Rocket Booster (SRB) MDMs*
7. *Engine Interface Units (EIU)*
8. *Payload Data Interleaver (PDI)*
9. *Payload Signal Processor (PSP)*
10. *Space Lab (SL) Experiment/Subsystem Computers*
11. Display Electronic Units (DEU)

(SLS - 2.1.1.3) The RTPS shall provide the capability, in Direct Input/Output (DIO) Mode (i.e., when the GPCs are not active), to issue commands to and receive measurements from:

1. *Solid Rocket Booster MDMs*
2. *Command Decoder MDMs*

(SLS - 2.1.1.4) The RTPS shall provide the capability to interface with all GPC Functional Destinations available via the LDB.~~, including:~~

1. Systems Software Avionics Command Support (SACS)
2. Test Control Supervisor Single Commands (TCS-1)
3. *Mass Memory (MM)/*Display Electronics Unit (DEU) Read
4. *Space Shuttle Main Engine (SSME) Load Program (SLP)*
5. *Launch Sequence (LS)*
6. *Test Control Supervisor Test Sequences (TCS-S)*
7. *Continuation of previous TCS-S sequence containing the same transaction ID*

(SLS - 2.1.1.5) *The RTPS shall provide the capability ~~(that is currently provided in LPS by theto load, modify, dump, and verify the Orbiter Computational Facilities (OCF))~~ to load, modify, dump, and verify the memory of Space Shuttle computers according to the following matrix:*

Function	**GPC Main Memory (PASS and BFS)**	**Mass Memory**	**DEU Memory**	**PCMMU**	**SSME**
Load		*X*	~~*X*~~		*X*
Dump	*X*	*X*	*X*	*X*	*X*
Verify	*X*	*X*	*X*	*X*	*X*
Modify	X	X	~~X~~		X

Table 2 GPC Main Memory Load, Modify, Dump, and Verify Commands

(SLS - 2.1.1.5.1) *The RTPS shall provide the capability to Uplink commands to the Orbiter GPC via the PCM Uplink interface as defined in:*

1. *SS-P-0002-140, Space Shuttle Downlist/Uplink Software Requirements.*
2. *MG038101, Backup System Services Program Requirements Document.*

(SLS - 2.1.1.5.2) *The RTPS Gateway shall be capable of interfacing to a minimum of three Uplink PCM data destinations (hardline, and two RF destinations) simultaneously, with only one of the sources being actively transmitted.*

(SLS - 2.1.1.5.3) *At CITE, the RTPS shall provide the capability to generate Uplink commands in the "NSP Switched Out" format.*

(SLS - 2.1.1.5.4) *At CITE, the RTPS shall provide the capability to generate Uplink commands in the "NSP Switched In" format.*

(SLS - 2.2.3.3) Command Processing
The CLCS must provide the capability to issue commands available with the LPS. In addition, the CLCS must issue new command types which will be required by systems that the CLCS will interface with in the future. (*Partial)*

(SLS - 2.2.3.3.1) The CLCS shall provide the capability to issue all keyboard commands in Appendix B~~C~~ with a Y in the column titled "IMPL" (implement). *(See Appendix)*

(SLS - 2.2.3.3.2) The CLCS shall provide the capability to protect from inadvertent issuance of commands.

1. Prerequisite logic
2. Two step protocol required on ~~critical~~ commands entered from the keyboard (e.g., arm, execute, and disarm logic)
3. Two step protocol required on commands issued using a pointing device.

(SLS - 2.2.3.3.3) The CLCS shall provide the capability to restrict issuance of commands to only authorized users and applications.

(SLS - 2.2.4.1) Manual Command, Monitoring and Control
The CLCS provides a means of manually commanding, monitoring, and controlling end item hardware and software similar to the Command Processor in the CCMS. The Command Processor functions will be analyzed on a one for one basis and each function will be allocated to the appropriate area within the CLCS for development.

(SLS - 2.2.4.1.1) The CLCS shall provide, using a Graphical User Interface (GUI) paradigm, the capabilities identified in Appendix B~~C~~ with a Y in the column titled "IMPL" (implement).

(SLS - 2.2.4.3.2) The RTPS shall provide a FD Viewer which provides a mechanism for viewing all available information about any RTPS ~~measurement~~ FD.

(SLS - 2.2.5.5) Prerequisite Control Logic
To protect against inadvertently issuing a command which could injure personnel or damage equipment, CLCS provides the capability for the user to predefine logic that verifies the appropriate conditions are met prior to executing the command.

(SLS - 2.2.5.5.1) RTPS shall provide the capability to verify that certain conditions are met before issuing any FD command to an End-Item.

(SLS - 2.2.5.5.3) RTPS shall provide the capability to ~~manually~~ override Prerequisite Control Logic prior to and ~~once~~ after a command ~~issuance~~ has been blocked.

(SLS - 2.2.5.5.4) RTPS shall provide the capability for Reactive Sequence Test applications to bypass Prerequisite Control Logic.

(SLS - 2.2.5.6.5) ***RTPS shall provide the capability for an End-Item Manager Test Application to issue a command to another End-Item Manager Test Application***. **(CORBA)**

(SLS - 2.2.5.6.6) *RTPS shall provide the capability to alter the state of frame rate/time domain control based on an event.*

(SLS - 2.2.5.7.5) Test Application Scripts shall be capable of issuing commands to End-Item Managers.

(SLS - 2.2.5.7.6) Test Application Scripts shall be capable of initiating other Test Application Scripts.

(SLS - 2.2.5.7.9) Test Application Scripts shall provide the capability to accept direct engineering input to confirm completion of manual operations.

(SLS - 2.2.5.8.5) CLCS shall provide the capability for users to define display applications that provide user manual End-Item control and visibility.

(SLS - 2.2.5.8.7) The RTPS User Display function shall provide the capability to issue commands to FDs by cursor control regardless of whether there is a End-Item Manager controlling the FDs or not. ~~cursor control commands to FDs not managed by End-Item Managers.~~

(SLS - 2.2.5.8.8) The RTPS User Display function shall provide the capability to issue cursor control commands to End-Item Managers.

(SLS - 2.2.5.8.9) The RTPS User Display function shall provide the capability to initiate Test Application Scripts.

(SLS - 2.2.10.1.6) The RTPS shall record all commands and command responses to the SDC.

1.4 Commanding and Command Processor Hardware Diagram

Not Applicable.

1.5 Commanding and Command Processor Thread Deliverables

The following is a list of deliverable products for the Commanding and Command Processor Thread

Deliverable	R&D Document	Code	API Manual	Users Guide
Command Management	x	x		
Command Processor	x	x		x
Command Scriptor	x	x		x
Viewers		x		x
Authentication Database	x			
Command GUI Viewer	x	x		
Command Interface GUI White Paper	x			
HMF OMS/RCS	x	x		
Application Services	x	x	x	

Table 3 Commanding and Command Processor Thread Deliverables

1.6 Commanding and Command Processor Thread Assessment Summary

This section contains the summary of the costs and labor involved in implementing the Commanding and Command Processor. It is broken into three sections. The first is a summary of the individual CI (CSCI and HWCI) labor assessments. The second is a summary of hardware costs. The third is a summary of procurement activities needed.

1.6.1 Labor Assessments

The total Labor Costs required to provide this Commanding and Command Processor Thread are summarized in the following table:

No.	CSCI/HWCI Name	Thor LM	Changes covered in
1	Command Support CSCI - Command Management CSC	13	Commanding and Command Processor Thread
2	Command Support CSCI - Command Processing CSC	14	Commanding and Command Processor Thread
3	System Services		
4	Application Services	TBD	Commanding and Command Processor Thread
5	System Viewers	3.5	Commanding and Command Processor Thread

No.	CSCI/HWCI Name	Thor LM	Changes covered in
6	Gateway Services		Gateway Interfaces Thread
7	TCID Build & Control		Gateway Interfaces Thread
8	Data Bank		Gateway Interfaces Thread
9	Test Build (Authentication DB)	TBD	Commanding and Command Processor Thread
10	Validation TCID Build	TBD	Commanding and Command Processor Thread
11	Math Model for Valid. TCID	TBD	
	TOTAL	30.5+	

Table 4 Commanding and Command Processor Thread Labor Assessments

1.7 Commanding and Command Processor Thread Schedule & Dependencies

1.7.1 Schedule

This section contains a schedule of major activities and milestones. Required activities and dates in the schedule are Phase One, Phase Two, Phase Three, Implementation, Unit Test, and Test. All CIs are expected to go through panels together. Add additional lines for exceptions. In the following matrix Start and Stop dates should be provided for activities. Milestones only require a finish date.

Task Name	**Start**	**Finish**
Thor Assessment Kickoff		7/23/97
Concept Panel Internal Review		9/24/97
Concept Panel		9/26/97
Thor Development		
Requirement Panel Internal Review		10/21/97
Requirement Panel		10/23/97
Design Panel Internal Review		11/11/97
Design Panel		11/13/97
CSCI Unit Testing	12/17/97	1/23/98
CSCI Development Integration Test	1/21/98	2/6/98
CSCI Formal Integration Test	2/23/98	3/6/98
Support System Integration Test	3/6/98	3/27/98
Thor Development Complete		3/27/98

Table 5 Commanding and Command Processor Thread Schedule

1.7.2 Dependencies

Table 6 identifies dependencies that the Commanding and Command Processor thread has in order to be satisfactorily specified, designed, implemented, and tested.

No.	Dependency Area	Dependency	Need Date
1		Simulation hardware connections	10/15/97
2		Thor Database updates	
3		Validation TCID	
4	Application Services	API definitions; object definitions	TBD
5	System Message Services	System Message APIs	12/1/97

No.	Dependency Area	Dependency	Need Date
6	System Services	Updated IPC (RM)	Dec/Jan.*
7	OPS CM	User Class Assignment to CC WS	

Table 6 Commanding and Command Processor Thread Dependencies

* System Services updated IPC (RM) is scheduled to be delivered on 1/9/98. Data Distribution has a dependency date of 12/1/97.

1.8 Commanding and Command Processor Thread Simulation Requirements

This section contains a list of Simulation Requirements needed to test the Commanding and Command Processor.

Math Models providing PCM Downlink data, LDB read/write functions, GSE functions, ME data
Physical link connections to KATS lab (available 10/97 Mike Harkins)
Physical links to SGOS/VSI (available 10/97 Mike Harkins)
SDE, IDE 1 & 2

1.9 Commanding and Command Processor Thread Integration and System Test

The Command and Command Processor Thread testing shall be accomplished on the SDE and IDE systems while attached to the LPS SGOS. A validation TCID and SGOS math model will be developed to provide a set of well known function designators and a predictable test environment. Command scripts will be developed to automate the testing process. The KATS lab will be used to perform commands such as LDB bus switching, GMEM Read and Write.

The TCID and SGOS model will support:
1. Launch Data Bus commands and measurements
2. GSE commands and measurements
3. PCM downlink measurements

Measurement type supported are:
1. Analog
2. Discrete
3. Digital pattern

Command types supported are:
1. Analog stimulus
2. Discrete stimulus
3. Digital Pattern Stimulus

1.10 Commanding and Command Processor Training Requirements

None.

1.11 Commanding and Command Processor Thread Facilities Requirements

None.

1.12 Travel Requirements

None.

1.13 Commanding and Command Processor Action Items/Resolution

None.

2. CSCI Assessments

This section provides the individual CSCI assessments. This information is summarized in section 1

2.1 Command Support CSC

Command Support CSCI will provide additional command syntax, command interfacing, enhance command flow, and support authentication and PCL. Command Support will provide a prototype Command Scripting GUI..

Command Management (CMM) Work Required

1. Divide CMM into two processes.
2. Add PCL interfacing
3. Enhance message routing.
4. Add authentication interfacing.
5. Interface with new IPC
6. Add additional System Messages as necessary.
7. Develop routing tables for multiple CCPs.

Command Processor (CMP) Work Required

1. Add syntax for PCL override.
2. Add syntax for the commands listed in Appendix A.
3. Add syntax for Timer commands.
4. Add syntax for LDB priority queue.
5. Add System Message Service API to retrieve system messages to be displayed in the GUI.

FD Commanding + other commanding interface (FDC +) Work Required

1. Add interfaces for additional objects related to commands listed in Appendix A.
2. Add Timer Commanding interface.

Command Scriptor Work Required

1. Design and prototype user interface.
2. Integrate the scripting interface with Command Processing Parse and Build objects.

Authentication Work Required

1. Develop the authentication overall design.
2. Provide the initial capabilities of authentication

PCL Work Required

1. Develop the PCL interface design.
2. Provide the initial interface to PCL

CSCI Assessment

CSC Name	CSC Labor (LM)	% of CSC
Command Management	4	25
Command Processor	10.5	10
FD Commanding	7~~3~~	10
Command Scriptor	3.5	40

CSC Name	CSC Labor (LM)	% of CSC
Authentication Interface	3	50
PCL Interface	3	50

Table 7 Command Support CSCI Assessments

Basis of estimate

- Command Management is assessed 4 MM to update its processing, add authentication and PCL interfaces, and add the updated SMS.
- Command Processor development is expected @1700 SLOC (8.5 MM) for new commands, 100 SLOC (.5mm) for SMS API, and 1.5 MM to add/update the SMS messages.
- FD Commanding development is estimated at @3000 SLOC. It is expected that new FD Commanding APIs will be based on already developed code (reuse) and although the SLOC is high the actual development time will be low. Thor introduces new objects for commanding (not FD). Additional code required.
- Command Scriptor prototype is a GUI development utilizing a GUI builder to create the user interface. Command Scriptor will reuse the Parse/Build code from Command Processor. The code that will require development is the functional interface.
- Authentication Interface will define the interface, develop an IDC, and the API for Authentication.
- PCL Interface will define the interface, develop an IDC, and the API for PCL processing.

Documentation

Provide your assessment of the kinds and amount of documentation that must be provided with the Commanding and Command Support CSCI.

Document Type	New/Update	Number of Pages
Requirements and Design Documentation	Update	@55 Pages
Users Guide: Command Processor	Update	@120~~50~~ Pages (Syntax and System Messages)
Users Guide: Command Scriptor	New	@10 Pages (Syntax and System Messages)
Authentication API Interface Document	New	@30 pages
FD CMDing Interface Design Document	Update	@20 pages
Test Procedure	New	@90 pages

Table 8 Command Support Documentation

Assumptions

None.

Open Issues

Blocking vs non blocking commands
Is there more than 1 CCP active in any Thor Configuration?

2.2 Application Services Assessment

This section provides an overview description of the changes necessary to Application Services (ASV) CSCI.

Function Designator Services Work Required

The following work will be performed during Thor :.

- Provide changes to FD object structure:
    - PCL sequence number (or identifier)
    - (possibly) PCL status indicator
    - ownership information (RSYS or equivalent)

This work includes interface with TCID/OLDB necessary to support PCL and ownership.

- Provide capability to pass along PCL “override”.

LDB Services Work Required

- Provide LDB objects and object class
    - provide any specific class structure needed for LDB priority queue

Gateway Interface Services Work Required

- Provide Gateway objects and object class
  This is to support activate/inhibit command, hardware change commands, etc.

Other Services (CSC title(s) TBD) Required

- . Provide identification services to support authentication
    - a “who am I” API
    - TBD
- Provide authorization query services [Note that this API does not do “authentication”. It provides an API for user-level programs to query their privileges. We don’t envision Command using this API; rather Command providing the API.]

ASV System Message Services Work Required

- Provide System Message Service API to retrieve system message text to be displayed. [Labor impact covered under System messages Thread.]

CSCI Assessment

CSC Name	CSC Labor (LM)	% of CSC
FD Services		
LDB Services	TBD	
Gateway Interface Services	TBD	
CSC title(s) TBD	TBD	
System Message Services	[4.0]	

Table 9 Application Services CSCI Assessment

Basis of estimate

Labor for ASV System Message Services has been included under the System Messages Enhancement Thread. Other labor estimates will be provided shortly, as soon as preliminary designs have been worked out.

Documentation

Commanding and Command Processor thread documentation is listed in Table 10.

Document Type	New/Update	Number of Pages

Document Type	New/Update	Number of Pages
Requirements and Design Documentation	Yes - ASV DP2 & 3	25
Users Guide	N/A	
API Interface Document	Yes - web format	TBD
Interface Design Document	Stub to web pages	
Test Procedure	Yes	80

Table 10 Application Services Documentation

Assumptions

Communications about commands for gateways and LDB will be handled in a similar fashion to FDs. Applications Services will not provide the actual coding that does authentication; ASV will provide the interface for this code. Test Build will provide all data related to user class assignments and FD-related information (such as PCL sequence assignments). Test Build will provide via flat files any information necessary for establishing class and object initializations..

Open Issues

Preliminary design is currently TBD. Expect it to be established within 2 weeks.
See open issues related to Command Management and Command Processor CSCs..

2.3 Test Build Assessment

This is an overview description of the changes necessary to Test Build to support the Thor Commanding and Command Processor Thread.

CSC Work Required

Provide authentication Data Base.

CSCI Assessment

CSC Name	CSC Labor (LM)	% of CSC
Authentication Data Base	TBD	

Table 11 Test Build Assessment

Basis of estimate

TBD

Documentation

Documentation provided with the Commanding and Command Processor thread.

Document Type	New/Update	Number of Pages
Requirements and Design Documentation		
Users Guide		
API Interface Document		

Document Type	New/Update	Number of Pages
Interface Design Document		
Test Procedure		

Table 12 Test Build Documentation

Assumptions

TBD

Open Issues

Authentication Definition.

2.4 System Viewer Assessment

Provide support evaluating command interfaces. A set of GUI prototypes will be developed and shown to users. These GUIs will be designed based on functionality from current and other command human interfaces. A white paper describing the lessons learned will be delivered.

Command GUI Viewer Work Required

This is a list of work to be accomplished for this function.

- Learn functionality of other command human interfaces.
- Design a set of command GUI prototypes and receive user feedback.
- White paper describing lessons learned.

CSC Name	CSC Labor (LM)	% of CSC
Command GUI Viewer	3.5 LM	

Table 13 System Viewer Assessment

Basis of estimate

Five weeks for understanding each command interface. (CCMS, TCMS, Launch Team Training System).
Two weeks for white paper.
One week interface with users.
Three weeks for command GUIs
Three weeks for command support and integration.

Documentation

Document Type	New/Update	Number of Pages
Command Interface GUI white paper	New	TBD

Table 14 System Viewer Documentation

Assumptions.

Open Issues

None.

2.5 Hypergolic Maintenance Facility OMS/RCS CSCI Assessment

The HMF OMS/RCS CSCI will provide a subset of software that will allow functional testing of the Real Time Control (RTC) Applications Software interfaces with the Command Management CSCI. This subset will consist of an SL-GMS display monitor and an End Item Manager (EIM) that can be used to test inter-application communications and manual command (cursor control) capabilities. Prerequisite Control Logic (PCL) sequences will also be provided.

2.5.1 Work Required

The work being performed in support of this thread includes development of an SL-GMS display monitor, an End Item Manager and associated PCL sequences for the Forward Reactive Control System (FRCS) check-out bay power subsystem. The work is part of the normal RTC Applications Software development effort for the HMF CSCI (special development to support Command Management testing will be kept to a minimum). CSUs from the HMF CSCI that will be made available for Command Management testing will be:

- FRCS Power Display Monitor(s)
- FRCS Power Subsystem EIM
- Miscellaneous PCL Sequences
- HMF OMS/RCS CSCI Management Software

This software will be added to the Thor Baseline delivery, but will not be validated for GSE/vehicle testing at this time.

2.5.2 Assessment

The following is the labor assessment for development of this software. It is provided here for reference only since the effort has already been included in the cost to support the HMF CSCI development.

CSC/CSU	*LaborMonths*	*% CSCI*
SL-GMS Display	0.25	5%
EIM	5.00	15%
PCL Sequences	0.50	5%
CSCI Management Software	1.00	50%

Table 15 HMF Assessment

2.5.3 Documentation

None. All documentation for the HMF OMS/RCS CSCI will be provided when the CSCI is delivered for vehicle/GSE testing.

3. HWCI Assessments

Not applicable.

3.1 HWCI Name_1 Assessment

N/A

Work Required

N/A

HWCI Assessment

N/A

4. COTS Products Dependencies

4.1 SW Products Dependency List

Provide a list of COTS SW needs.

Product Name	Quantity Needed	Need Date

4.2 HW Products Dependency List

None.

APPENDIX A. — CCMS KEYBOARD

Command Portability Matrix

The tables on the following pages list existing CCMS commands, OC tasks, Transients, and Programs. These commands and programs were compiled by searching the existing CCMS Design Documents and User's Guides for each. The tables on the following pages also contain reference data for each command and program indicating where additional data can be found for each in the existing documentation. Following is a description of each column in the tables:

- COMMAND DESCRIPTION - A description of the command and the proposed syntax.
- PORT - N/Y = Whether existing command should be recoded.
  - E = Equivalent functionality required
  - D = Duplicate functionality provided elsewhere
  - SDC = SDC should provide this functionality
- 2-S - Whether command is a 2-step command.
  (S = Sometimes, under some conditions)
- SRC - Valid sources for this command:
  - M = Master
  - I = Integration
  - B = Backup
  - A= all,
  - R = Responsible System
  - R* = Responsible System (R) can only A/I FD functions, not entire Gateway functions (such as data acquisition)
  - L = Local System
  - M* = Master and any other CPU that the Master allows via the A KU (Allow Keyboard Utilities) or similar command.
  - MM = Master Master
  - SDC = Shuttle Data Center
- CP - Command Processor command
- SV - System Viewer command
- SA - System Application (See Glossary)
- UA - User Application (See Glossary)
- AP - Executable by User-Generated Applications
- M - Method - L=List, P=Pulldown, G=Graphic, C=Command Line

COMMAND DESCRIPTION	PORT	2-S	SRC	CP	SV	SA	UA	AP-METHOD
STATUS FUNCTION DESIGNATORS (033-1, Pkg 18)								
Status Function Designator (S FD)	Y		Any	Y	Y			Y
SYSTEM STATUS (033-1, Pkg 20)								
System Status (SYST)	Y		MIB*			Y		
ACTIVATE/INHIBIT COMMANDS (033-1, Pkg 22)								
Act/Inh Data Acquisition (A/I DA)	Y	S	MIR	Y				Y
Act/Inh Data Processing (A/I PR)	Y	S	MIR	Y				Y
Act/Inh Command Issuance (A/I CM)	Y	Y	MIR	Y				Y
Act/Inh HIM Polling Command (A/I HI)	Y	Y	MIB	Y				
Act/Inh HIM Testing Command (A/I HT)	Y	S	MI	Y				
Activate Static Dump								
CHANGE COMMANDS (033-1, Pkg 23) (Table Maintenance)								
Change EU Coefficients (C EUC)	E	Y	MIRB	Y				Y
Change Bit Sync Error Count (C PSB)	Y	Y	MIB	Y				Y
Change/Select PCM (PCMS)	Y	Y	MIB	Y				Y
Change Hardware Address (C HA)	Y	Y	MIRB	Y				
Change Sample Rate (C RA)	Y	Y	MIRB	Y				Y
APPLY/ISSUE/SET COMMANDS (033-1, Pkg 24)								
Apply Analog (APPL)	Y	Y	MIR	Y				Y
Set Discrete (SET)	Y	Y	MIR	Y				Y
Issue Digital Pattern (ISSU)	Y	Y	MIR	Y				Y
GPC KEYBOARD INTERFACE TCS COMMANDS (033-1, Pkg 26)								
GMEM Read (GPCC LDBA, or GPCC LDBD)	Y		All				Y	Y
GMEM Write (GPCC LDBA, GPCC LDBD, or GPCC UPLK)	Y		DPS				Y	Y
LDB Control (GPCC LDBC)	Y		DPS				Y	Y
DEU AND ONBOARD COMPUTER COMMUNICATIONS COMMANDS (033-1, Pkg. 27)								
Read Onboard Values (READ)	Y						Y	Y
Display Electronic Unit (DEUE) Equivalent	Y	Y		Y				Y
NEW COMMANDS								
Read GSE HIM Output Function Designator (R FD)	New		Any	Y	Y			Y
Read EU Polynomial (R EUP)	New		Any	Y	Y			
Switch Launch Data Busses (SW LDB)	Y	Y	MI	Y				Y

THIS IS THE LAST PAGE OF THE DOCUMENT